Claims

We claim:

1. A computer-implemented method for programmatically creating a graphical program, comprising:

creating a first graphical program, wherein the first graphical program includes a graphical program creation node for programmatically creating a new graphical program, wherein the first graphical program also includes at least one object creation node for programmatically creating at least one graphical program object in the new graphical program;

executing the first graphical program;

creating the new graphical program in response to said executing the first graphical program; and

including the at least one graphical program object in the new graphical program in response to said executing the first graphical program.

2. The method of claim 1, wherein said executing the first graphical program includes:

executing the graphical program creation node, wherein said executing the graphical program creation node causes creation of the new graphical program;

executing the object creation node, wherein said executing the object creation node causes inclusion of the graphical program object in the new graphical program.

3. The method of claim 1, wherein the first graphical program further includes a property node; the method further comprising:

the property node getting or setting a property of the graphical program object in response to said executing the first graphical program.

4. The method of claim 3, wherein the object creation node outputs a reference to the graphical program object;

wherein the property node receives as input the reference to the graphical program object;

wherein the property node gets or sets a property of the graphical program object specified by the reference to the graphical program object.

5. The method of claim 1, wherein the first graphical program further includes an invoke node; the method further comprising:

the invoke node invoking a method on the graphical program object in response to said executing the first graphical program.

6. The method of claim 5, wherein the object creation node outputs a reference to the graphical program object;

wherein the invoke node receives as input the reference to the graphical program object;

wherein the invoke node invokes a method on the graphical program object specified by the reference to the graphical program object.

7. The method of claim 5, wherein the invoked method connects the graphical program object to another graphical program object in the new graphical program.

8. The method of claim 7, wherein said connecting the graphical program object to said another graphical program object comprises connecting an input of the graphical program object to an output of said another graphical program object.

9. The method of claim 5, wherein the invoked method performs one of 1) moving the graphical program object to another location in the new graphical program; and 2) resizing the graphical program object in the new graphical program.

10. The method of claim 1, wherein the first graphical program includes a plurality of object creation nodes each for programmatically creating a graphical program object in the new graphical program, wherein said plurality of object creation nodes includes a first object creation node for creating a first graphical program object in the new graphical program and includes a second object creation node for creating a second graphical program object in the new graphical program;

wherein said including the at least one graphical program object in the new graphical program comprises including the first graphical program object and the second graphical program object in the new graphical program;

wherein the first graphical program further includes an invoke node;

wherein said executing the first graphical program includes executing the invoke node, wherein said executing the invoke node connects the first graphical program object to the second graphical program object.

11. The method of claim 1, wherein said creating the first graphical program comprises:

displaying the graphical program creation node;

displaying the object creation node;

configuring the object creation node with one or more inputs.

12. The method of claim 11, wherein the graphical program creation node outputs a reference to the new graphical program;

wherein said connecting the graphical program creation node to the object creation node includes connecting the reference to the new graphical program to an input of the object creation node.

13. The method of claim 11, further comprising configuring the graphical program creation node with one or more inputs, wherein said configuring the graphical program creation node with one or more inputs comprises performing one or more of:

1) specifying a new graphical program type for the graphical program creation node; 2) specifying a template graphical program for the graphical program creation node; and 3) specifying a server reference for the graphical program creation node.

14. The method of claim 13, wherein a server reference is specified for the graphical program creation node;

wherein said executing the first graphical program comprises executing program instructions on a first computer;

wherein the server reference references a server program running on a second computer;

wherein the second computer is connected to the first computer via a network;

wherein said creating the new graphical program in response to said executing the first graphical program comprises the server program creating the new graphical program.

15. The method of claim 11, wherein said configuring the object creation node with one or more inputs comprises performing one or more of:

1) specifying an object class for the object creation node; 2) specifying an object sub-class for the object creation node; 3) specifying position information to the object creation node; and 4) specifying owner reference information for the object creation node.

16. The method of claim 1, wherein said creating the first graphical program comprises:

displaying the graphical program creation node;

specifying a new graphical program type for the graphical program creation node;

wherein said creating the new graphical program comprises creating the new graphical program of the specified new graphical program type.

17. The method of claim 16, wherein the graphical program creation node includes a type input;

wherein said specifying a new graphical program type for the graphical program creation node comprises connecting type information to the type input of the graphical program creation node.

18. The method of claim 1, wherein said creating the first graphical program comprises:

displaying the graphical program creation node;

specifying a template graphical program for the graphical program creation node;

wherein said creating the new graphical program comprises creating the new graphical program based on the template graphical program.

19. The method of claim 18, wherein the graphical program creation node includes a template input;

wherein said specifying a template graphical program for the graphical program creation node comprises connecting information specifying an existing graphical program to the template input of the graphical program creation node.

20. The method of claim 1, wherein said creating the first graphical program comprises:

displaying the graphical program creation node;

specifying a reference to a server program for the graphical program creation node;

wherein said creating the new graphical program comprises the server program creating the new graphical program.

21. The method of claim 20, wherein the server program is an application instance of a graphical programming environment.

22. The method of claim 20, wherein the graphical program creation node includes a server program reference input;

wherein said specifying a reference to a server program for the graphical program creation node comprises connecting information specifying a server program to the server program reference input of the graphical program creation node.

23. The method of claim 20, wherein said executing the first graphical program is performed in a first computing system;

wherein said server program executes in a second computing system;

wherein the first computing system is connected to the second computing system.

24. The method of claim 1, wherein said creating the first graphical program comprises:

displaying the object creation node;

specifying a graphical program object class for the object creation node;

wherein the graphical program object included in the new graphical program is of the graphical program object class.

25. The method of claim 24, wherein said creating the first graphical program further comprises:

specifying a graphical program object sub-class for the object creation node;

wherein the graphical program object included in the new graphical program is of the object sub-class.

26. The method of claim 1, wherein said creating the first graphical program comprises:

displaying the object creation node;

specifying position information to the object creation node;

wherein the graphical program object included in the new graphical program is positioned in the new graphical program at a location based on the position information.

27. The method of claim 1, wherein said creating the first graphical program comprises:

displaying the object creation node;

specifying owner reference information for the object creation node, wherein the owner reference information designates an owner entity;

wherein the graphical program object is included in the new graphical program as a member of the owner entity.

28. The method of claim 27, wherein the owner entity is an entity from the group consisting of: 1) the new graphical program and 2) another graphical program object of the new graphical program.

29. The method of claim 1, wherein the new graphical program includes a block diagram, wherein the graphical program object is a function node placed in the block diagram.

30. The method of claim 1, wherein the new graphical program includes a block diagram, wherein the graphical program object is a programmatic structure placed in the block diagram.

31. The method of claim 1, wherein the new graphical program includes a user interface panel, wherein the graphical program object is a user interface object placed in the user interface panel.

32. The method of claim 31, wherein the user interface object is a user interface input object placed in the user interface panel for viewing input to the new graphical program.

33. The method of claim 31, wherein the user interface object is a user interface input object placed in the user interface panel for providing input to the new graphical program.

34. The method of claim 31, wherein the user interface object is a user interface output object placed in the user interface panel for viewing output of the new graphical program.

35. The method of claim 31, wherein the new graphical program also includes a block diagram, wherein the user interface object is a user interface input object placed in the user interface panel for viewing input to the block diagram.

36. The method of claim 31, wherein the new graphical program also includes a block diagram, wherein the user interface object is a user interface output object placed in the user interface panel for viewing output from the block diagram.

37. The method of claim 1, wherein said executing the first graphical program occurs in a first computing environment;

wherein said first computing environment is connected to a second computing environment;

wherein said executing the first graphical program comprises sending information from the first computing environment to the second computing environment;

wherein the new graphical program is created in the second computing environment.

38. A computer-implemented method for programmatically creating a graphical program, comprising:

creating a first graphical program, wherein the first graphical program includes a graphical program creation node for programmatically creating a new graphical program, wherein the first graphical program also includes a first object creation node for creating a first object in the new graphical program and includes a second object creation node for creating a second object in the new graphical program, wherein the first graphical program further includes an invoke node;

executing the first graphical program;

creating the new graphical program in response to said executing the first graphical program;

including the first object and the second object in the new graphical program in response to said executing the first graphical program; and

the invoke node connecting the first object to the second object in response to said executing the first graphical program.

39. The method of claim 38, wherein creating the first graphical program comprises:

configuring the invoke node to invoke a connect method, wherein the connect method is operable to connect two graphical program objects;

providing references to the first object and the second object as inputs to the invoke node.

40. The method of claim 38, wherein said connecting the first object to the second object comprises connecting an output of the first object to an input of the second object.

41. A computer-implemented method for programmatically accessing a graphical program, comprising:

creating a first graphical program, wherein the first graphical program includes a reference to an existing graphical program, wherein the first graphical program also includes a property node or an invoke node, wherein the property node is operable to get or set a property of the existing graphical program or a property of an object of the existing graphical program, and the invoke node is operable to invoke a method on the existing graphical program or on an object of the existing graphical program;

executing the first graphical program;

accessing the existing graphical program in response to said executing the first graphical program.

42. The method of claim 41, wherein the first graphical program includes a property node;

wherein the reference to the existing graphical program is provided to the property node;

wherein the property node is configured to get or set a property of the existing graphical program.

43. The method of claim 41, wherein the first graphical program also includes a node for obtaining a graphical program object reference;

wherein the reference to the existing graphical program is provided to the node for obtaining a graphical program object reference;

wherein the node for obtaining a graphical program object reference is configured to obtain a reference to a particular object of the existing graphical program;

wherein the first graphical program includes a property node;

wherein the reference to the particular object of the existing graphical program is provided to the property node;

wherein the property node is configured to get or set a property of the particular object of the existing graphical program.

44. The method of claim 41, wherein the first graphical program includes an invoke node;

wherein the reference to the existing graphical program is provided to the invoke node;

wherein the invoke node is configured to invoke a particular method on the existing graphical program.

45. The method of claim 41, wherein the first graphical program also includes a node for obtaining a graphical program object reference;

wherein the reference to the existing graphical program is provided to the node for obtaining a graphical program object reference;

wherein the node for obtaining a graphical program object reference is configured to obtain a reference to a particular object of the existing graphical program;

wherein the first graphical program includes an invoke node;

wherein the reference to the particular object of the existing graphical program is provided to the invoke node;

wherein the invoke node is configured to invoke a particular method on the particular object of the existing graphical program.

46. A computer-implemented method for programmatically creating a graphical program, comprising:

creating a first program, wherein the first program includes a graphical program creation function for programmatically creating a new graphical program, wherein the first program also includes an object creation function for programmatically including an object in the new graphical program;

executing the first program;

creating the new graphical program in response to said executing the first program; and

including the object in the new graphical program in response to said executing the first program.

47. The method of claim 46, wherein the first program is a graphical program;

wherein the graphical program creation function comprises a graphical program creation node;

wherein the object creation function comprises an object creation node.

48. The method of claim 46, wherein the first program is a text-based program.

49. The method of claim 48, wherein the graphical program creation function comprises a method call to create the new graphical program;

wherein the object creation function comprises a method call to create the object.

50. The method of claim 48, wherein the text-based program obtains a reference to a software component;

wherein the software component enables the text-based program to perform the method call to create the new graphical program;

wherein the software component enables the text-based program to perform the method call to create the object.

51. The method of claim 50, wherein the software component interfaces with a server program;

wherein the server program receives the method call to create the new graphical program;

wherein the server program creates the new graphical program;

wherein the server program receives the method call to create the object;

wherein the server program creates the object.

52. The method of claim 50, wherein the software component is an ActiveX component.

53. A computer-implemented method for programmatically accessing a graphical program, comprising:

creating a first program, wherein the first program includes a function for obtaining a reference to an existing graphical program, wherein the first program also includes a property function or an invoke function, wherein the property function gets or sets a property of the existing graphical program or an object of the existing graphical program, and the invoke function invokes a method on the existing graphical program or an object of the existing graphical program;

executing the first program;

creating the new graphical program in response to said executing the first program; and

including the object in the new graphical program in response to said executing the first program.

54. A system for programmatically creating or accessing a graphical program, comprising:

a computer system including a CPU and memory;

a client program executing in the computer system, wherein the client program performs API calls to programmatically create or access a graphical program;

a server program operable to receive the client program calls to programmatically create or access a graphical program and operable to perform the respective operations.

55. The system of claim 54, wherein the server program executes an another computer system, wherein said another computer system is connected to said computer system via a network.

56. The system of claim 54, wherein the client program performs said calls to programmatically create or access a graphical program by obtaining a reference to a software component and invoking methods of the software component;

wherein the software component is operable to perform the operations of programmatically creating or accessing the graphical program.

57. The system of claim 54, wherein the client program performs said calls to programmatically create or access a graphical program by obtaining a reference to a software component and invoking methods of the software component;

wherein the software component relays the client program calls to the server program.

58. The system of claim 54, wherein the server program is a graphical programming environment application.

59. The system of claim 54, wherein the client program is a client graphical program;

wherein the client graphical program includes a graphical program creation node for programmatically creating a new graphical program;

wherein the client graphical program also includes an object creation node for programmatically creating a graphical program object in the new graphical program;

wherein said API calls to programmatically create or access a graphical program comprise calls resulting from executing the graphical program creation node and the object creation node.

60. The system of claim 59, wherein the client graphical program further includes a property node for getting or setting a property of the graphical program object.

61. The system of claim 59, wherein the client graphical program further includes an invoke node for invoking a method on the graphical program object.

62. The system of claim 61, wherein the object creation node is a first object creation node for programmatically creating a first graphical program object in the new graphical program;

wherein the graphical program also includes a second object creation node for programmatically creating a second graphical program object in the new graphical program;

wherein the invoked method connects the first graphical program object to the second graphical program object.

63. A memory medium comprising program instructions executable to implement:

creating a first graphical program, wherein the first graphical program includes a graphical program creation node for programmatically creating a new graphical program, wherein the first graphical program also includes at least one object creation node for programmatically creating at least one graphical program object in the new graphical program;

executing the first graphical program;

creating the new graphical program in response to said executing the first graphical program; and

including the at least one graphical program object in the new graphical program in response to said executing the first graphical program.

64. A client program for programmatically creating a new graphical program, wherein the client program comprises:

a means for instantiating the new graphical program;

a means for adding an object to the new graphical program;

a means for getting or setting properties of the new graphical program or the object;

a means for invoking methods on the new graphical program or the object.

65. The client program of claim 64, wherein the client program is a graphical program;

wherein said means for instantiating the new graphical program comprises a graphical program creation node;

wherein said means for adding an object to the new graphical program comprises an object creation node;

wherein said means for getting or setting properties of the new graphical program or the object comprises a property node;

wherein said means for invoking methods on the new graphical program or the object comprises an invoke node.

66. A client program for programmatically accessing a graphical program, wherein the client program comprises:

a means for obtaining a reference to the graphical program;

a means for obtaining a reference to a particular ~~object of~~ the graphical program;

a means for getting or setting properties of the graphical program or the object;

a means for invoking methods on the graphical program or the object.

ADD A1

Add B3